

Goal: Practise semaphore usage.

Marks: 50

Due: February 28, 2018

Description: (Please read the complete assignment document before starting.)

A professor, TA and student are responsible for planting seeds. The student digs the holes, the professor places a seed in the empty hole, and the TA fills the hole with the seed. Several synchronization constraints apply to this activity:

1. The professor can plant a seed only if at least one empty hole exists.
2. The TA cannot fill a hole unless at least one unfilled hole exists into which a seed has been placed.
3. The professor has asked the TA to keep an eye on the student, so the student cannot get more than MAX holes ahead of the TA. If there are MAX unfilled holes, the student has to wait.
4. There is only one shovel with which both the student and TA need to dig and fill the holes, respectively.

Your goal is to simulate the work being completed by the professor, TA, and student. Using the Java Semaphore Class, complete the *Planting.java* code provided to synchronize the student, TA, and Professor threads according to the above synchronization rules.

- In addition to the Semaphore class constructor, you may use the following Semaphore methods for synchronization: *acquire()*, *acquireUninterruptible()*, *tryAcquire()*, and *release()*.
- **Your solution should only use semaphores (using the Semaphore class), not Java synchronization primitives (i.e. you should not use synchronized, wait(), and notify()).**
- The simulation is started by launching “*java Planting*”.
- An example of the output is provided (see *log.txt*).

To submit your solution:

Submit single file *Planting.java*, containing your solution. You will get partial marks for partial solutions. Be sure to include your name and student number at the start of the file.